

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Jun KAMADA, et al.

Application No.:

Group Art Unit:

Filed: January 11, 2002

Examiner:



For: CODE EXECUTION APPARATUS AND CODE DISTRIBUTING METHOD

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Assistant Commissioner for Patents
Washington, D.C. 20231

#2
J. Cuthbert
2/26/02

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith a certified copy of the following foreign application:

Japanese Patent Application No. 2001-162271

Filed: May 30, 2001

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing date(s) as evidenced by the certified papers attached hereto, in accordance with the requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: January 11, 2002

By: _____

James D. Halsey, Jr.
Registration No. 22,729

700 11th Street, N.W., Ste. 500
Washington, D.C. 20001
(202) 434-1500

JAPAN PATENT OFFICE



This is to certify that the annexed is a true copy of the following application as filed with this Office.

Date of Application: May 30, 2001

Application Number: Patent Application
No. 2001-162271

Applicant(s): FUJITSU LIMITED

October 2, 2001

Commissioner,
Japan Patent Office Kozo OIKAWA

Certificate No. 2001-3089998

日 本 国 特 許 庁
JAPAN PATENT OFFICE

J1017 U.S. PTO
10/042262
01/11/02

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 5月30日

出 願 番 号

Application Number:

特願2001-162271

出 願 人

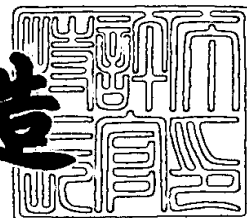
Applicant(s):

富士通株式会社

2001年10月 2日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3089998

【書類名】 特許願

【整理番号】 0150536

【提出日】 平成13年 5月30日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 15/16

【発明の名称】 コード実行装置およびコード配布方法

【請求項の数】 10

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 蒲田 順

【発明者】

 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社内

 【氏名】 小谷 誠剛

【特許出願人】

 【識別番号】 000005223

 【氏名又は名称】 富士通株式会社

【代理人】

 【識別番号】 100074099

 【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F

 【弁理士】

 【氏名又は名称】 大菅 義之

 【電話番号】 03-3238-0031

【選任した代理人】

 【識別番号】 100067987

 【住所又は居所】 神奈川県横浜市鶴見区北寺尾7-25-28-503

 【弁理士】

 【氏名又は名称】 久木元 彰

【電話番号】 045-573-3683

【手数料の表示】

【予納台帳番号】 012542

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9705047

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コード実行装置およびコード配布方法

【特許請求の範囲】

【請求項 1】 マルチプロセッサシステムを用いたコード実行装置であって

セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検証するための検証用情報とを格納するセキュアメモリ手段と、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗号化コードを実行するセキュアプロセッサ手段と、

通常タスクのコードを格納する通常メモリ手段と、

前記通常タスクのコードを実行する通常プロセッサ手段と、

前記セキュアタスクと通常タスクの振り分けを行って、前記暗号化コードを前記セキュアメモリ手段に格納し、前記通常タスクのコードを前記通常メモリ手段に格納する制御手段と

を備えることを特徴とするコード実行装置。

【請求項 2】 前記セキュアメモリ手段は、物理メモリ割り当ての単位毎に前記暗号化コードを格納し、該単位毎の暗号化コードに対する検証用情報を格納し、該検証用情報を用いて該単位毎の暗号化コードを検証し、前記セキュアプロセッサ手段は、正当であると検証された暗号化コードに含まれる暗号化命令をフェッチし、復号して実行することを特徴とする請求項 1 記載のコード実行装置。

【請求項 3】 前記暗号化コードを固有鍵でさらに暗号化して格納するセキュアドライブ手段をさらに備え、該セキュアドライブ手段と前記セキュアメモリ手段は、相互認証を行ってセッション鍵を共有し、該セキュアドライブ手段は、前記制御手段からの読み出し指示に基づいて、該暗号化コードを該固有鍵で復号し、該セッション鍵で暗号化して、該セキュアメモリ手段に転送することを特徴とする請求項 1 記載のコード実行装置。

【請求項 4】 前記セキュアプロセッサ手段は、前記暗号化コードを用いて、該暗号化コードを実行する論理回路の少なくとも一部を、回路状態として不揮発的に固定することを特徴とする請求項 1 記載のコード実行装置。

【請求項 5】 物理メモリ割り当ての単位毎に暗号化コードを格納する手段と、

該単位毎の暗号化コードが正当であることを検証するための検証用情報を格納する手段と、

該検証用情報を用いて該単位毎の暗号化コードを検証する手段とを備えることを特徴とするメモリ。

【請求項 6】 暗号化コードを格納するメモリから、該暗号化コードが正当であることを示す通知を受け取る手段と、

前記通知を受け取ったとき、前記暗号化コードに含まれる暗号化命令をフェッチして復号する手段と、

復号された命令を実行する手段とを備えることを特徴とするプロセッサ。

【請求項 7】 コンピュータのためのプログラムを記録した記録媒体であって、該プログラムは、

セキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含むマルチプロセッサシステムにおいて、該セキュアタスクと通常タスクの振り分けを行い、

前記セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検証するための検証用情報とを、セキュアメモリに格納し、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗号化コードをセキュアプロセッサに実行させる

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 8】 セキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含むマルチプロセッサシステムにおいて、該セキュアタスクと通常タスクの振り分けを行い、

前記セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検証するための検証用情報とを、セキュアメモリに格納し、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗

号化コードをセキュアプロセッサに実行させる

処理をコンピュータに実行させるためのプログラム。

【請求項 9】 コード作成者が、コード認証機関に実行可能なコードを提供し、

前記コード認証機関が、前記コードが正当であることを検証するための検証用情報を該コードに付加して、マルチプロセッサシステムのユーザに配布し、

前記マルチプロセッサシステムは、前記コードを用いてセキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含み、該セキュアタスクと通常タスクの振り分けを行い、前記検証用情報を用いて該コードが正当であることを検証し、該コードを実行する

ことを特徴とするコード配布方法。

【請求項 10】 コード作成者が、コード認証機関に実行可能なコードを提供して、手数料を支払い、

前記コード認証機関が、前記コードが正当であることを検証するための検証用情報を該コードに付加し、

前記コード作成者が、前記コードをマルチプロセッサシステムのユーザに配布して、該ユーザが支払う対価を受領し、

前記マルチプロセッサシステムは、前記コードを用いてセキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含み、該セキュアタスクと通常タスクの振り分けを行い、前記検証用情報を用いて該コードが正当であることを検証し、該コードを実行する

ことを特徴とするコード配布方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、コード化されたプログラムのような、実行可能なコードを実行する装置およびそのようなコードを配布する方法に関する。

【0002】

【従来の技術】

電子署名と暗号化が施された実行可能なコード（以下では、単にコードと記す）を、署名を検証した上で復号しながら実行することで、認証されたコードのみが安全に動作する環境が実現できる。実行可能なコードには、コード化されたプログラムの一部または全体が含まれる。このような動作環境の実現方法として、署名検証機能と復号機能を持つプロセッサ（セキュアプロセッサ）をメモリマップドの入出力装置（I/O装置）として見せかける方法が考えられる。この方法では、暗号化コードをデータとしてそのI/O装置に渡し、実行させて、実行結果を取得する。

【0003】

【発明が解決しようとする課題】

しかしながら、上述した従来のコード実行方法には、次のような問題がある。

このコード実行方法では、I/O装置のメモリ量の制限からサイズの大きなコード全体を一度に渡すことができない。また、一旦コードを渡して実行を開始すると、終わるまで他の制御を行うことができないため、署名検証や復号等のセキュア処理をマルチタスクで実行することができない。このため、セキュア処理を伴う複数のタスクを効率よく実行することができない。

【0004】

このうち、後者の問題は、I/O装置をマルチタスク処理が可能なように設計し、そのI/O装置専用のタスク管理モジュールをオペレーティングシステム（OS）内に用意することで解決できる。しかし、OSが、I/O装置（セキュアプロセッサ）専用のタスク管理モジュールを、通常プロセッサ用のタスク管理モジュールと重複して持つことになり、効率上好ましくない。

【0005】

さらに、セキュリティ上の観点から、OSそのものがセキュアプロセッサ上で動作することが望ましいが、既存OSをいきなりセキュアプロセッサ用に書き換えることは簡単ではないという問題もある。

【0006】

本発明の課題は、既存OSを大幅に変更することなく、電子署名と暗号化が施されたコードを効率よく実行する装置と、そのような装置に対してコードを配布

する方法を提供することである。

【 0 0 0 7 】

【課題を解決するための手段】

図 1 は、本発明のコード実行装置の原理図である。図 1 のコード実行装置は、マルチプロセッサシステムを用いて実現され、セキュアメモリ手段 1 1、セキュアプロセッサ手段 1 2、通常メモリ手段 1 3、通常プロセッサ手段 1 4、および制御手段 1 5 を備える。

【 0 0 0 8 】

セキュアメモリ手段 1 1 は、セキュアタスクの暗号化コードと、その暗号化コードが正当であることを検証するための検証用情報とを格納する。セキュアプロセッサ手段 1 2 は、検証用情報により暗号化コードが正当であると検証されたとき、その暗号化コードを実行する。通常メモリ手段 1 3 は、通常タスクのコードを格納し、通常プロセッサ手段 1 4 は、通常タスクのコードを実行する。

【 0 0 0 9 】

制御手段 1 5 は、セキュアタスクと通常タスクの振り分けを行って、暗号化コードをセキュアメモリ手段 1 1 に格納し、通常タスクのコードを通常メモリ手段 1 3 に格納する。

【 0 0 1 0 】

検証用情報としては、例えば、電子署名、パリティコード、CRC (Cyclic Redundancy Check) ビットが用いられる。また、制御手段 1 5 は、例えば、マルチプロセッサシステムの OS に対応する。

【 0 0 1 1 】

セキュアタスクを実行する場合、制御手段 1 5 は、その暗号化コードを検証用情報とともにセキュアメモリ手段 1 1 に格納する。そして、検証用情報を用いて暗号化コードが正当であると検証されると、セキュアプロセッサ手段 1 2 が、その暗号化コードを実行する。

【 0 0 1 2 】

また、通常タスクを実行する場合、制御手段 1 5 は、そのコードを通常メモリ手段 1 3 に格納する。そして、通常プロセッサ手段 1 4 は、そのコードを実行す

る。

【 0 0 1 3 】

このように、マルチプロセッサシステム内でセキュアタスクと通常タスクの共存を許し、制御手段 1 5 がそれらのタスクをセキュアプロセッサ手段 1 2 と通常プロセッサ手段 1 4 に振り分けることで、タスク管理が簡単になる。したがって、OS を大幅に変更することなく、セキュアタスクのコードを効率よく実行することができる。

【 0 0 1 4 】

また、このようなマルチプロセッサシステムに対してコードを配布する方法としては、例えば、以下の 2 通りの方法がある。

(1) コード作成者が、コード認証機関に実行可能なコードを提供し、コード認証機関が、コードが正当であることを検証するための検証用情報をコードに付加して、マルチプロセッサシステムのユーザに配布する。

(2) コード作成者が、コード認証機関に実行可能なコードを提供して、手数料を支払い、コード認証機関が、検証用情報をコードに付加する。その後、コード作成者が、コードをマルチプロセッサシステムのユーザに配布して、ユーザが支払う対価を受領する。

【 0 0 1 5 】

【発明の実施の形態】

以下、図面を参照しながら、本発明の実施の形態を詳細に説明する。

本実施形態では、マルチプロセッサシステム中の 1 つのプロセッサをセキュアプロセッサに置き換え、ヘテロなマルチプロセッサシステムを構築する。そして、セキュアなタスクと非セキュアなタスクの各プロセッサへの割り当てを、OS が制御する。このように、セキュアプロセッサを含むヘテロなマルチプロセッサシステムを構築することで、OS が 2 種類のタスク管理モジュールを重複して持つ必要がなくなる。

【 0 0 1 6 】

また、セキュアタスクと非セキュアタスクの共存を許すことで、可能な部分からセキュアタスク化していき、最終的に OS の全タスクをセキュア化するという

マイグレーションが可能となる。このようなマイグレーションは、OSが細かなタスクの集合体として実現されているような場合に限られるが、これにより、既存OSをいきなりセキュアプロセッサ用書き換える必要がなくなる。

【0017】

しかし、このようなシステムにおいては、セキュアプロセッサが暗号化コードを1命令ずつ取り出して（フェッチして）実行するため、コード全体をセキュアプロセッサにまとめて渡すことはできない。したがって、暗号化コード全体に署名をつけたのでは、署名の検証ができないという問題が発生する。そこで、本実施形態では、メモリ割り当ての最小単位であるページ（例えば、4 Kバイト等）単位に署名を付与した形で暗号化コードを作成し、メモリ割り当て時にメモリ自身が署名を検証する構成を採用する。

【0018】

さらに、このシステムに対して配布すべきコードを収集し、収集したコードに署名をつけて配布する機関を設ける。これにより、コード作成者は、コードを広範囲に配布することが可能となり、ユーザは、安心してコードを利用することが可能となる。

【0019】

図2は、このようなマルチプロセッサシステムの構成図である。図2のシステムは、通常メモリ21、通常プロセッサ22、セキュアメモリ23、セキュアプロセッサ24、およびセキュアドライブ／メディア25を含む。これらの装置はシステムバス26により互いに接続されているが、通常プロセッサ22は、セキュアメモリ23から命令をフェッチすることではなく、セキュアプロセッサ24は、通常メモリ21から命令をフェッチすることはない。

【0020】

通常プロセッサ22は、通常メモリ21を用いて通常のタスク（非セキュアタスク）の通常のコードを実行し、セキュアプロセッサ24は、セキュアメモリ23を用いてセキュアタスクの暗号化コードを実行する。セキュアドライブ／メディア25は、セキュアタスク用の暗号化コードを格納する格納装置である。図2では、通常プロセッサ22とセキュアプロセッサ24が1つずつ設けられている

が、各プロセッサを複数設けてもよい。

【 0 0 2 1 】

図 3 は、セキュアメモリ 2 3 とセキュアプロセッサ 2 4 の構成例を示している。図 3 のセキュアメモリ 2 3 は、認証局公開鍵 3 1、署名検証部 3 2、署名保持部 3 3、ページ 3 4 を有する。ページ 3 4 は、物理メモリ（実メモリ）割り当ての最小単位であり、例えば、4 K バイトの容量を持つ。また、署名保持部 3 3 は、ページ毎に、ページに対する署名データを格納する領域を持つ。署名検証部 3 2 は、例えば、ハードウェアまたは M P U（Micro Processing Unit）を用いて実装され、ページ単位で署名を用いて暗号化コードを検証する。

【 0 0 2 2 】

署名としては、例えば、認証局（Certificate Authority，CA）の秘密鍵によって作成され、セキュアメモリ 2 3 にあらかじめ格納された CA の公開鍵によって検証可能な X. 5 0 9 証明書が用いられる。

【 0 0 2 3 】

セキュアプロセッサ 2 4 は、復号鍵設定部 4 1、復号鍵保持部 4 2、復号部 4 3、およびプロセッサ 4 4 を有する。このうち、復号鍵設定部 4 1、復号鍵保持部 4 2、および復号部 4 3 は、命令を実行するプロセッサ 4 4 の命令入力部の前段に設けられ、復号部 4 3 は、例えば、ハードウェアまたは M P U を用いて実装される。

【 0 0 2 4 】

署名を含む暗号化コードは、セキュアドライブ／メディア 2 5 より読み出され、署名と暗号化コードに分離された後、それぞれ署名保持部 3 3 とページ 3 4 に格納される。

【 0 0 2 5 】

これらの格納が終了すると、署名検証部 3 2 は、署名を作成した認証局の公開鍵で署名を検証し、問題がなければ、署名に含まれる暗号化コードのハッシュ値を、ページ 3 4 上の暗号化コードから再計算したハッシュ値と比較する。

【 0 0 2 6 】

これらのハッシュ値が一致し、暗号化コードが改竄されていないことを確認す

ると、ページ 3 4 上の暗号化コードが有効である（正当なコードである）ことをセキュアプロセッサ 2 4 に通知する。また、ハッシュ値が一致しなければ、ページ 3 4 上の暗号化コードが無効である（正当なコードでない）ことをセキュアプロセッサ 2 4 に通知する。

【 0 0 2 7 】

セキュアプロセッサ 2 4 の復号部 4 3 は、有効の通知を受け取ると、ページ 3 4 上のメモリアドレスから必要な暗号化命令をフェッチし、復号鍵保持部 4 2 の復号鍵を用いて暗号化命令を順次復号する。そして、プロセッサ 4 4 は、復号された命令を順次実行する。復号に必要な復号鍵は、復号鍵設定部 4 1 により、あらかじめ復号鍵保持部 4 2 に設定されている。

【 0 0 2 8 】

さらに、セキュアプロセッサ 2 4 に複数の復号鍵を格納しておき、どの復号鍵を使用して復号するかを外部より指定可能とすることもできる。図 4 は、このようなセキュアプロセッサ 2 4 の構成図である。

【 0 0 2 9 】

図 4 のセキュアプロセッサ 2 4 は、複数の復号鍵保持部 4 2 を持つ点と、これらの保持部 4 2 に格納された複数の復号鍵のうちのどれを使用するかを指示する復号鍵指示部 4 5 が追加されている点が、図 3 のセキュアプロセッサ 2 4 とは異なる。復号鍵指示部 4 5 は、例えば、ハードウェアまたは M P U を用いて実装される。どの復号鍵を使用するかは、例えば、現在実行中のセキュアタスクに応じて、O S が復号鍵指示部 4 5 に指示する。

【 0 0 3 0 】

図 5 は、図 2 のマルチプロセッサシステムの動作を制御するセキュア O S の構成図である。図 5 のセキュア O S 5 1 は、セキュアプロセッサ 2 4 または／および通常プロセッサ 2 2 上で動作し、セキュアタスク管理 5 2、セキュアメモリ管理 5 3、およびセキュアファイルシステム 5 4 を含む。

【 0 0 3 1 】

セキュアタスク管理 5 2 とセキュアメモリ管理 5 3 は、セキュアタスクと非セキュアタスクの振り分けを行う。これにより、セキュアタスクの暗号化コードは

セキュアメモリ 2 3 に格納され、非セキュアタスクのコードは通常メモリ 2 1 に格納される。また、セキュアタスク管理 5 2 は、セキュアタスクと非セキュアタスクの両方のマルチタスク処理を制御する。以下、対象とするタスクがセキュアタスクの場合の動作を説明する。

【 0 0 3 2 】

セキュアタスク管理 5 2 は、複数のタスクのコンテキストを管理し、コンテキスト切り替え時に、セキュアプロセッサ 2 4 のプログラムカウンタの変更等の通常処理のほか、セキュアプロセッサ 2 4 内に保持されている復号鍵のうちどれを使用するかを指示する。

【 0 0 3 3 】

セキュアメモリ管理 5 3 は、必要に応じて、セキュアタスクへのセキュアメモリ 2 3 の割り当てを行う。なお、セキュアメモリ 2 3 からセキュアプロセッサ 2 4 への暗号化命令の転送は、CPU (Central Processing Unit) のフェッチ動作であるので、セキュアメモリ管理 5 3 を経由しない。

【 0 0 3 4 】

セキュアファイルシステム 3 4 は、セキュアドライブ／メディア 2 5 に格納された暗号化コードのファイルを管理する。そして、セキュアメモリ管理 5 3 からの要求に応じて、セキュアドライブ／メディア 2 5 から暗号化コードを読み出し、セキュアメモリ管理 5 3 に渡す。

【 0 0 3 5 】

次に、図 6 から図 9 までを参照しながら、セキュアタスク管理 5 2、セキュアメモリ管理 5 3、およびセキュアファイルシステム 5 4 の処理をより詳細に説明する。

【 0 0 3 6 】

図 6 は、セキュアタスク管理 5 2 の処理のフローチャートである。図 6 の処理は、セキュアプロセッサ 2 4 上で現在実行中のセキュアタスクのタイムスライスが切れ、タイマ割り込みが発生して、セキュアタスク管理 5 2 に制御が移ったときに開始される。

【 0 0 3 7 】

セキュアタスク管理 5 2 は、まず、スケジューリングアルゴリズムに従って、次に実行するセキュアタスク A を決定し（ステップ S 1）、セキュアタスク A のコンテキストを復元する（ステップ S 2）。このとき、セキュアプロセッサ 2 4 のプログラムカウンタおよびスタックポインタの復元や、セキュアプロセッサ 2 4 とセキュアメモリ 2 3 の間に存在する MMU（Memory Management Unit）内の TLB（Translation Look aside Buffer）の復元等が行われる。

【 0 0 3 8 】

次に、プログラム復号鍵としてセキュアタスク A 用のものを使用することを、セキュアプロセッサ 2 4 に指示する（ステップ S 3）。そして、セキュアタスク A のタイムスライス（例えば、1 0 0 m s）をタイマに設定し（ステップ S 4）、セキュアプロセッサの動作を再開する（ステップ S 5）。

【 0 0 3 9 】

図 7 は、セキュアメモリ管理 5 3 の処理のフローチャートである。図 7 の処理は、セキュアタスク実行中にページフォルトが起き、割り込みが発生して、セキュアメモリ管理 5 3 に制御が移ったときに開始される。

【 0 0 4 0 】

セキュアメモリ管理 5 3 は、まず、セキュアメモリ 2 3 内に未使用の実メモリ領域があるか否かをチェックし（ステップ S 1 1）、実メモリがあれば、それを 1 ページ割り当てる（ステップ S 1 3）。未使用の実メモリがない場合は、実メモリ解放処理のサブルーチンを呼び出して（ステップ S 1 2）、空きを作った後に実メモリを割り当てる。

【 0 0 4 1 】

次に、割り当てた実メモリのアドレスと仮想アドレスの対応表を作成し、MMU 内の TLB に格納する（ステップ S 1 4）。そして、割り当てた実メモリに配置するコードをセキュアファイルシステム 5 4 に要求し、受け取ったコードを実メモリに配置して（ステップ S 1 5）、セキュアプロセッサ 2 4 の動作を再開する（ステップ S 1 6）。

【 0 0 4 2 】

図 8 は、図 7 のステップ S 1 2 において呼び出されたサブルーチンが行う実メ

メモリ解放処理のフローチャートである。サブルーチンは、まず、実メモリ解放アルゴリズムに従って、ページアウトの対象となる実メモリを決定する（ステップ S 2 1）。次に、対象となった実メモリ上のコードを、セキュアドライブ／メディア 2 5 にページアウトする（書き出す）（ステップ S 2 2）。そして、呼び出し元に復帰する（ステップ S 2 3）。

【 0 0 4 3 】

図 9 は、セキュアファイルシステム 5 4 の処理のフローチャートである。図 9 の処理は、図 7 のステップ S 1 5 においてセキュアメモリ管理 5 3 からコードの要求を受けたときに開始される。

【 0 0 4 4 】

セキュアファイルシステム 5 4 は、まず、対象となるプログラムの先頭からのオフセットを受け取り（ステップ S 3 1）、セキュアドライブ／メディア 2 5 内で指定位置までシークする（ステップ S 3 2）。そして、指定位置より 1 ページ分のコードを読み出し、セキュアメモリ管理 5 3 に渡す（ステップ S 3 3）。

【 0 0 4 5 】

ところで、セキュアメモリ 2 3 とセキュアプロセッサ 2 4 が相互認証を行ってセッション鍵を共有することで、暗号化コードをより安全にやり取りすることも可能である。この場合、セキュアメモリ 2 3 は、暗号化コードをセッション鍵でさらに暗号化した上でセキュアプロセッサ 2 4 に転送する。

【 0 0 4 6 】

図 1 0 は、このようなセキュアメモリ 2 3 とセキュアプロセッサ 2 4 の構成図である。図 1 0 のセキュアメモリ 2 3 は、図 3 の構成に加えて、さらに相互認証／セッション鍵共有部 6 1 および暗号部 6 2 を有し、セキュアプロセッサ 2 4 は、図 4 の構成に加えて、さらに相互認証／セッション鍵共有部 7 1 および復号部 7 2 を有する。

【 0 0 4 7 】

まず、相互認証／セッション鍵共有部 6 1 と相互認証／セッション鍵共有部 7 1 は、相互に信頼できる相手であることを認証した上で、セッション鍵を生成／共有する。相互認証の方法は、公開鍵を用いた証明書ベースの方法でもかまわな

いし、共通鍵を用いた方法でもかまわない。また、セッション鍵は、例えば、乱数を用いて生成される。

【 0 0 4 8 】

その後、セキュアメモリ 2 3 の暗号部 6 2 は、ページ 3 4 上の暗号化命令をセッション鍵でさらに暗号化して、セキュアプロセッサ 2 4 に転送する。セキュアプロセッサ 2 4 の復号部 7 2 は、受け取った暗号化命令をセッション鍵で復号した後、復号部 4 3 に渡す。その後、図 4 に示したように、暗号化命令が対応する復号鍵で復号されて実行される。

【 0 0 4 9 】

また、同様にして、セキュアドライブ／メディア 2 5 とセキュアメモリ 2 3 が相互認証を行ってセッション鍵を共有することで、暗号化コードをより安全にやり取りすることも可能である。

【 0 0 5 0 】

図 1 1 は、このようなセキュアドライブ／メディア 2 5 とセキュアメモリ 2 3 の構成図である。図 1 1 のセキュアドライブ／メディア 2 5 は、格納媒体 8 1、格納装置固有鍵 8 2、相互認証／セッション鍵共有部 8 3、復号部 8 4、および暗号部 8 5 を有し、セキュアメモリ 2 3 は、図 1 0 の構成に加えて、さらに復号部 6 3 を有する。

【 0 0 5 1 】

セキュアドライブ／メディア 2 5 は、暗号化コードを格納装置固有鍵 8 2 または格納媒体固有鍵 8 6 でさらに暗号化して、格納媒体 8 1 に格納する。格納媒体 8 1 としては、磁気ディスク、光ディスク、光磁気ディスク、磁気テープ等が用いられる。また、格納装置固有鍵 8 2 は、セキュアドライブ／メディア 2 5 固有の鍵であり、格納媒体固有鍵 8 6 は、格納媒体 8 1 固有の鍵である。

【 0 0 5 2 】

相互認証／セッション鍵共有部 8 3 と相互認証／セッション鍵共有部 6 1 は、図 1 0 の場合と同様にして、相互に信頼できる相手であることを認証した上で、セッション鍵を生成／共有する。

【 0 0 5 3 】

セキュアドライブ／メディア 2 5 の復号部 8 4 は、格納媒体 8 1 上に格納された暗号化コード 8 7 を、格納装置固有鍵 8 2 または格納媒体固有鍵 8 6 で復号して、暗号部 8 5 に渡す。暗号部 8 5 は、相互認証／セッション鍵共有部 8 3 が保持しているセッション鍵により暗号化コードをさらに暗号化して、セキュアメモリ 2 3 に転送する。セキュアメモリ 2 3 の復号部 6 3 は、受け取った暗号化コードをセッション鍵で復号して、元の暗号化コードの形に戻した上で、ページ 3 4 に格納する。

【 0 0 5 4 】

このとき、図 5 のセキュアファイルシステム 5 4 は、セキュアドライブ／メディア 2 5 とセキュアメモリ 2 3 の間のセッション鍵の共有を仲介する。その後、セキュアファイルシステム 5 4 は、このセッション鍵によって暗号化された暗号化コードを、格納媒体 8 1 上の論理フォーマットに従ってセキュアドライブ／メディア 2 5 から読み出し、セキュアメモリ 2 3 に転送する。

【 0 0 5 5 】

ここで、1 つのセキュアタスクが実行される場合の処理の流れを説明する。この場合、セキュアタスクには現在セキュアメモリが 1 ページだけ割り当てられており、プログラムカウンタはそのページ上の暗号化コードの最後の暗号化命令を指しているものとする。また、各エンティティ（セキュアメモリ 2 3、セキュアプロセッサ 2 4、セキュアドライブ／メディア 2 5）間の相互認証処理、セッション鍵の共有処理、セッション鍵による暗号化／復号処理については、説明を省略する。

（1）セキュアプロセッサ 2 4 は、セキュアメモリ 2 3 より暗号化コードをフェッチし、復号した後に実行する。

（2）セキュアプロセッサ 2 4 は、プログラムカウンタをインクリメントし、次の命令のフェッチ動作を実行する。

（3）実メモリが未割り当てであるため、セキュアメモリ 2 3 は、ページフォルト例外をセキュアタスク管理 5 2 に対して発生する。

（4）セキュアタスク管理 5 2 は、実行中のセキュアタスクをスリープ状態に設定した後、セキュアメモリ管理 5 3 に新たな実メモリの割り当てを依頼する。

(5) セキュアメモリ管理 5 3 は、新たな実メモリ 1 ページをセキュアタスクに割り当てる。

(6) セキュアタスク管理 5 2 は、暗号化コードの続きを読み出すように、セキュアファイルシステム 5 4 に依頼する。

(7) セキュアファイルシステム 5 4 は、暗号化コードの続きをセキュアドライブ／メディア 2 5 より読み出し、新たに割り当てられた実メモリに格納する。

(8) セキュアタスク管理 5 2 は、スリープ状態のセキュアタスクを実行状態に設定する。

(9) セキュアプロセッサ 2 4 は、新たに割り当てられたページ上の次の命令をフェッチして実行する。

【 0 0 5 6 】

次に、2つのセキュアタスク A、B が実行される場合の処理の流れを説明する。この場合、セキュアタスク A および B ともに、十分なセキュアメモリが割り当てられており、ページフォルトは起きないものとする。

(1) セキュアプロセッサ 2 4 は、セキュアタスク A の暗号化命令をフェッチして実行する。

(2) セキュアタスク管理 5 2 は、タイムスライスが切れ、タイマ割り込みが発生したため、セキュアタスク A をスリープ状態に設定する。

(3) セキュアタスク管理 5 2 は、スケジューリングアルゴリズムに従って、次に動作させるタスクをセキュアタスク B に決定し、セキュアタスク B を動作状態に設定する。

(4) セキュアタスク管理 5 2 は、セキュアタスク B の復号に必要な鍵をセキュアプロセッサ 2 4 に対して指示する。

(5) セキュアタスク管理 5 2 は、プログラムカウンタ、スタックポインタ、TLB のアドレス対応表等をセキュアタスク B 用に設定する。

(6) セキュアプロセッサ 2 4 は、セキュアタスク B の暗号化命令をフェッチして実行する。

【 0 0 5 7 】

以上の説明は、セキュア OS がセキュアプロセッサ 2 4 上で動作するものと考え

えると容易に理解できる。しかし、セキュアプロセッサ24の実行を一時停止する機能や、セキュアプロセッサ24のプログラムカウンタの変更等のようなコンテキストを切り替える機能がセキュアプロセッサ24に用意されていれば、セキュアOS自身は通常プロセッサ22上で動作させることも可能である。

【0058】

図2のマルチプロセッサシステムでは、セキュアメモリ23と通常メモリ21が別々に設けられているが、セキュアメモリ23と通常メモリ21の一部または全部がオーバーラップした形態も考えられる。

【0059】

図12および図13は、このようなマルチプロセッサシステムの構成例を示している。ただし、ここではセキュアドライブ／メディア25は省略されている。図12において、セキュアプロセッサ24と通常プロセッサ22は同一のシステムバス92（データバス、アドレスバス）を介して、セキュアメモリ91に接続されている。この場合、セキュアメモリ91は、図2のセキュアメモリ23と通常メモリ21の機能を兼ね備えている。

【0060】

また、図13において、セキュアプロセッサ24は、システムバス94を介して固有のセキュアメモリ23に接続されており、システムバス95を介して共有メモリ93に接続されている。また、通常プロセッサ22は、システムバス96を介して固有の通常メモリ21に接続されており、システムバス95を介して共有メモリ93に接続されている。共有メモリ93は、セキュアプロセッサ24と通常プロセッサ22に共通のメモリであり、セキュアメモリ23または／および通常メモリ21の機能を備えている。

【0061】

図12の構成は、システムバスおよびメモリが1つずつしかないため、図13の構成よりコストが低いという利点がある。しかし、セキュアプロセッサ24と通常プロセッサ22の両方がセキュアメモリ91にアクセス可能なため、図13の構成よりセキュリティレベルが低くなる。逆に、図13の構成は、図12の構成よりコストが増加するが、セキュリティレベルも向上する。

【 0 0 6 2 】

以上の実施形態では、セキュアプロセッサ 2 4 がコードをフェッチして実行しているが、コードの一部または全部を用いて、暗号化命令をフェッチし復号した上で実行する論理回路を自動的に生成することも考えられる。この場合、汎用論理回路を特定の回路状態に固定する装置がシステム内に設けられる。

【 0 0 6 3 】

セキュアメモリ 2 3 が正当なコードであることを検証した後、セキュアプロセッサ 2 4 は、そのコードを用いて論理回路の一部または全部を、回路状態として不揮発的に固定する。このとき、前の回路状態を消去して、新しく上書きする。

【 0 0 6 4 】

図 1 4 は、このような回路生成処理のフローチャートである。セキュアプロセッサ 2 4 は、まず、暗号化命令をフェッチして復号し（ステップ S 4 1）、コードを演算処理回路構成情報に翻訳する（ステップ S 4 2）。次に、回路構成情報を配線情報に翻訳し（ステップ S 4 3）、配線情報を不揮発的に焼き付ける（ステップ S 4 4）。配線情報の焼き付け方法としては、例えば、以下の 2 通りが考えられる。

（1）図 1 5 に示すように、複数の基本回路を配列（アレイ）状に並べ、配線情報に基づいて回路間を不揮発的に接続し、演算器を構成する。

（2）図 1 6 に示すように、構成済みの基本演算器を多種用意しておき、配線情報に基づいて必要な演算器間を不揮発的に接続する。

【 0 0 6 5 】

このように、処理部分をハードウェア化することで処理速度が向上する。また、ハードウェアとソフトウェア処理を併用すれば、暗号化命令を階層化して、セキュリティレベルを向上させることもできる。例えば、特に重要な部分の命令は、厳重な認証ステップを経てハードウェア化しておき、それ以外の命令は、ユーザの利便性を図るため、軽い認証で毎回ソフトウェア処理する。

【 0 0 6 6 】

以上の実施形態では、署名を用いてコードが正当か否かが検証されているが、コードが正当であることを検証するための情報（検証用情報）としては、他の任

意の情報を用いることができる。例えば、パリティコード、CRC (Cyclic Redundancy Check) ビット等を付加しておくことで、コードが壊れているか否かを検証することができる。そこで、以下では、署名の代わりに検証用情報という言葉を用い、この情報をコードに付加する機関をコード認証機関と呼ぶことにする。

【 0 0 6 7 】

次に、図 1 7 から図 2 2 までを参照しながら、検証用情報が付加されたコードの配布方法について説明する。

図 1 7 は、ユーザへのコード配布方法を示している。図 1 7 において、コード作成者 1 0 1 は、コード認証機関 1 0 2 にコードを提供する (P 1)。コード認証機関 1 0 2 は、受け取ったコードの正当性を確認した上で、検証用情報を付加し、認証済みコードをコード利用者 1 0 3 (ユーザ) に提供する (P 2)。コード利用者 1 0 3 は、例えば、上述したマルチプロセッサシステムを保有しており、受け取ったコードに付加された検証用情報を用いてコードの正当性を確認した後に、そのコードを利用する。

【 0 0 6 8 】

このとき、コード認証機関 1 0 2 は、コード作成者 1 0 1 に対して対価を提示してコードを収集し、収集に際して対価を支払う。そして、コード利用者 1 0 3 に対してコードの対価を提示し、検証用情報を付加した後、コード利用者 1 0 3 に対してコードを提供すると同時に対価を徴収する。

【 0 0 6 9 】

図 1 8 は、このような対価の支払いを示している。図 1 8 において、コード作成者 1 0 1 は、コード認証機関 1 0 2 にコードを提供し (P 1 1)、その対価をコード認証機関 1 0 2 から受け取る (P 1 2)。コード認証機関 1 0 2 は、コード利用者 1 0 3 に認証済みコードを提供し (P 1 3)、コード利用者 1 0 3 は、それに対する対価をコード認証機関 1 0 2 に支払う (P 1 4)。

【 0 0 7 0 】

コード利用者 1 0 3 およびコード認証機関 1 0 2 が支払う対価は、コード提供時に一度に課金されてもかまわないし、コードの利用／提供状況に応じて従量課

金されてもかまわない。後者の場合、例えば、コード利用者 1 0 3 が受領したコード数に応じて課金が行われる。

【 0 0 7 1 】

また、コード作成者 1 0 1 がコード認証機関 1 0 2 に手数料を支払うことで、コードに検証用情報を付加してもらい、コード利用者 1 0 3 が支払う対価を受領することも可能である。

【 0 0 7 2 】

図 1 9 は、このようなコード配布方法を示している。図 1 9 において、コード作成者 1 0 1 は、コード認証機関 1 0 2 にコードを提供し（P 2 1）、検証用情報を付加してもらうのに必要な手数料を支払って（P 2 2）、認証済みコードを取得する（P 2 3）。次に、認証済みコードをコード利用者 1 0 3 に提供して（P 2 4）、その対価を受け取る（P 2 5）。

【 0 0 7 3 】

コード利用者 1 0 3 が支払う対価は、コード提供時に一度に課金されてもかまわないし、コードの利用／提供状況に応じて従量課金されてもかまわない。また、コード作成者 1 0 1 が支払う手数料も、同様に、一括課金でもかまわないし、従量課金でもかまわない。

【 0 0 7 4 】

また、コード作成者 1 0 1 の代わりに、コード認証機関 1 0 2 がコードを配布してもよい。この場合、コード認証機関 1 0 2 が、認証済みコードをコード利用者 1 0 3 に提供して対価を徴収し、徴収した対価をコード作成者 1 0 1 に支払う。

【 0 0 7 5 】

また、図 1 7 のコード配布方法において、コード認証機関 1 0 2 は、受け取ったコードを 2 つ以上の部分に分割し、最初に一部を配布し、その後、コード利用者 1 0 3 の要求に応じて、残りの部分を配布することも可能である。この場合、最初の配布は、例えば、以下のいずれかの方法で行われる。

- （１）複数のユーザにコードを放送する。
- （２）各ユーザに、ネットワーク上からコードを自由にダウンロードさせる。

(3) コードを可搬記録媒体に収納して、その記録媒体をユーザに配布する。

【0076】

図20は、このようなコード配布方法を示している。図20において、コード作成者101は、コード認証機関102にコードを提供する(P31)。コード認証機関102は、コードの正当性を確認した後、検証用情報を付加して、コード利用者103に認証済みコードの一部を提供する(P32)。コード利用者103は、提供された一部のコードが正当であることを検証用情報を用いて確認した後、そのコードを利用する。さらに、必要であれば、残りの認証済みコードをコード認証機関102より入手して利用する(P33)。

【0077】

最初に提供されるコードは、例えば、印刷機能に制限を設けた年賀状作成ソフトや、最初の画面データのみを収録したゲームソフト等である。残りのコードは、例えば、すべての機能制限を取り除いた年賀状作成ソフトや、第2画面以降を収録したゲームソフト等である。

【0078】

このとき、コード認証機関102は、コード作成者101に対して対価を提示してコードを収集し、収集に際して対価を支払う。次に、コード利用者103に対して残しておいた部分のコードの対価を提示し、検証用情報を付加した後、コードを提供すると同時に対価を徴収する。

【0079】

図21は、このような対価の支払いを示している。図21において、P41およびP42の処理は、図18のP11およびP12の処理と同様である。次に、コード認証機関102は、認証済みコードの一部を、例えば、雑誌付録のCD-ROM(compact disk read only memory)やインターネットを通じて、無償で配布する(P43)。これを入手して利用したコード利用者103は、さらに残りのコードを利用したい場合には、コード認証機関102に対価を支払い(P45)、残りのコードの提供を受ける(P44)。

【0080】

また、図19と同様に、コード作成者101がコード認証機関102に手数料

を支払うことで、コードに検証用情報を付加してもらうことも可能である。この場合、コード作成者 1 0 1 がコード利用者 1 0 3 に対して残りの部分のコードの対価を提示し、コード利用者 1 0 3 に対してコードを提供すると同時に対価を徴収する。

【 0 0 8 1 】

図 2 2 は、このようなコード配布方法を示している。図 2 2 において、P 5 1、P 5 2、および P 5 3 の処理は、図 1 9 の P 2 1、P 2 2、および P 2 3 の処理と同様である。次に、コード作成者 1 0 1 は、認証済みコードの一部を、例えば、上述したような方法で無償配布する（P 5 4）。これを入手して利用したコード利用者 1 0 3 が残りのコードを利用したい場合には、対価を支払い（P 5 6）、残りのコードを入手する（P 5 5）。

【 0 0 8 2 】

また、コード作成者 1 0 1 の代わりに、コード認証機関 1 0 2 がコードを配布してもよい。この場合、コード認証機関 1 0 2 が、コード利用者 1 0 3 に対して残りの部分のコードの対価を提示し、コード利用者 1 0 3 に対してコードを提供すると同時に対価を徴収し、徴収した対価をコード作成者 1 0 1 に支払う。

【 0 0 8 3 】

以上説明したようなコード配布方法によれば、コード認証機関による認証済みのコードが配布されるので、ユーザは、安心してコードを利用することができる。これにより、コードを利用するユーザが増加し、コードを広範囲に配布することが可能となる。

【 0 0 8 4 】

ところで、図 5 のセキュア OS 5 1 は、例えば、セキュアドライブ／メディア 2 5 にあらかじめ格納され、必要に応じてメモリにロードされて、動作を開始する。また、セキュア OS 5 1 を外部に保存しておき、必要に応じてシステムにインストールすることも可能である。

【 0 0 8 5 】

図 2 3 は、セキュア OS 5 1 を含むプログラムとデータをマルチプロセッサシステムに供給することのできるコンピュータ読み取り可能な記録媒体を示してい

る。

【 0 0 8 6 】

サーバ 1 1 1 のデータベース 1 1 2 や可搬記録媒体 1 1 3 に保存されたプログラムとデータは、マルチプロセッサシステムのメモリ 1 1 4 にロードされる。このとき、サーバ 1 1 1 は、プログラムとデータを搬送する搬送信号を生成し、ネットワーク上の任意の伝送媒体を介してマルチプロセッサシステムに送信する。そして、マルチプロセッサシステムは、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【 0 0 8 7 】

可搬記録媒体 1 1 3 としては、メモ리카ード、フロッピーディスク、CD-R OM、光ディスク、光磁気ディスク等、任意のコンピュータ読み取り可能な記録媒体が用いられる。また、メモリ 1 1 4 は、図 2 の通常メモリ 2 1 またはセキュアメモリ 2 3、図 1 2 のセキュアメモリ 9 1、あるいは図 1 3 の共有メモリ 9 3 に対応する。

(付記 1) マルチプロセッサシステムを用いたコード実行装置であって、

セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検証するための検証用情報とを格納するセキュアメモリ手段と、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗号化コードを実行するセキュアプロセッサ手段と、

通常タスクのコードを格納する通常メモリ手段と、

前記通常タスクのコードを実行する通常プロセッサ手段と、

前記セキュアタスクと通常タスクの振り分けを行って、前記暗号化コードを前記セキュアメモリ手段に格納し、前記通常タスクのコードを前記通常メモリ手段に格納する制御手段と

を備えることを特徴とするコード実行装置。

(付記 2) 前記セキュアメモリ手段は、物理メモリ割り当ての単位毎に前記暗号化コードを格納し、該単位毎の暗号化コードに対する検証用情報を格納し、該検証用情報を用いて該単位毎の暗号化コードを検証し、前記セキュアプロセッサ手段は、正当であると検証された暗号化コードに含まれる暗号化命令をフェッチ

し、復号して実行することを特徴とする付記 1 記載のコード実行装置。

(付記 3) 前記セキュアプロセッサ手段は、複数の復号鍵を保持し、該複数の復号鍵のうち指示された復号鍵を用いて、前記暗号化命令を復号することを特徴とする付記 2 記載のコード実行装置。

(付記 4) 前記セキュアメモリ手段とセキュアプロセッサ手段は、相互認証を行ってセッション鍵を共有し、該セキュアメモリ手段は、前記暗号化命令を該セッション鍵でさらに暗号化して該セキュアプロセッサ手段に転送することを特徴とする付記 2 記載のコード実行装置。

(付記 5) 前記暗号化コードを固有鍵でさらに暗号化して格納するセキュアドライブ手段をさらに備え、該セキュアドライブ手段と前記セキュアメモリ手段は、相互認証を行ってセッション鍵を共有し、該セキュアドライブ手段は、前記制御手段からの読み出し指示に基づいて、該暗号化コードを該固有鍵で復号し、該セッション鍵で暗号化して、該セキュアメモリ手段に転送することを特徴とする付記 1 記載のコード実行装置。

(付記 6) 前記セキュアメモリ手段と通常メモリ手段の領域の少なくとも一部がオーバーラップしていることを特徴とする付記 1 記載のコード実行装置。

(付記 7) 前記セキュアプロセッサ手段は、前記暗号化コードを用いて、該暗号化コードを実行する論理回路の少なくとも一部を、回路状態として不揮発的に固定することを特徴とする付記 1 記載のコード実行装置。

(付記 8) 前記セキュアプロセッサ手段は、前記論理回路の前の回路状態を消去して、新しく上書きすることを特徴とする付記 7 記載のコード実行装置。

(付記 9) 物理メモリ割り当ての単位毎に暗号化コードを格納する手段と、
該単位毎の暗号化コードが正当であることを検証するための検証用情報を格納する手段と、

該検証用情報を用いて該単位毎の暗号化コードを検証する手段と
を備えることを特徴とするメモリ。

(付記 10) 暗号化コードを格納するメモリから、該暗号化コードが正当であることを示す通知を受け取る手段と、

前記通知を受け取ったとき、前記暗号化コードに含まれる暗号化命令をフェッ

チして復号する手段と、

復号された命令を実行する手段と
を備えることを特徴とするプロセッサ。

(付記 1 1) コンピュータのためのプログラムを記録した記録媒体であって、
該プログラムは、

セキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常
プロセッサを含むマルチプロセッサシステムにおいて、該セキュアタスクと通常
タスクの振り分けを行い、

前記セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検
証するための検証用情報とを、セキュアメモリに格納し、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗
号化コードをセキュアプロセッサに実行させる

処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り
可能な記録媒体。

(付記 1 2) セキュアタスクを実行するセキュアプロセッサと、通常タスクを
実行する通常プロセッサを含むマルチプロセッサシステムにおいて、該セキュア
タスクと通常タスクの振り分けを行い、

前記セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検
証するための検証用情報とを、セキュアメモリに格納し、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗
号化コードをセキュアプロセッサに実行させる

処理をコンピュータに実行させるためのプログラム。

(付記 1 3) コンピュータのためのプログラムを該コンピュータに搬送する搬
送信号であって、該プログラムは、

セキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常
プロセッサを含むマルチプロセッサシステムにおいて、該セキュアタスクと通常
タスクの振り分けを行い、

前記セキュアタスクの暗号化コードと、該暗号化コードが正当であることを検
証するための検証用情報とを、セキュアメモリに格納し、

前記検証用情報により前記暗号化コードが正当であると検証されたとき、該暗号化コードをセキュアプロセッサに実行させる

処理を前記コンピュータに実行させることを特徴とする搬送信号。

(付記 1 4) コード作成者が、コード認証機関に実行可能なコードを提供し、前記コード認証機関が、前記コードが正当であることを検証するための検証用情報を該コードに付加して、マルチプロセッサシステムのユーザに配布し、

前記マルチプロセッサシステムは、前記コードを用いてセキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含み、該セキュアタスクと通常タスクの振り分けを行い、前記検証用情報を用いて該コードが正当であることを検証し、該コードを実行する

ことを特徴とするコード配布方法。

(付記 1 5) 前記コード認証機関は、前記コード作成者に対して対価を提示して前記コードを収集し、収集に際して対価を支払い、前記ユーザに対して該コードの対価を提示し、前記検証用情報を付加した後、該ユーザに対して該コードを提供すると同時に対価を徴収することを特徴とする付記 1 4 記載のコード配布方法。

(付記 1 6) 前記コード認証機関は、前記コードを 2 つ以上の部分に分割し、最初に一部を配布し、その後、前記ユーザの要求に応じて、残りの部分を配布することを特徴とする付記 1 4 記載のコード配布方法。

(付記 1 7) 前記コード認証機関は、前記コード作成者に対して対価を提示してコードを収集し、収集に際して対価を支払い、前記ユーザに対して前記残りの部分の対価を提示し、検証用情報を付加した後、コードを提供して対価を受領することを特徴とする付記 1 6 記載のコード配布方法。

(付記 1 8) コード作成者が、コード認証機関に実行可能なコードを提供して、手数料を支払い、

前記コード認証機関が、前記コードが正当であることを検証するための検証用情報を該コードに付加し、

前記コード作成者が、前記コードをマルチプロセッサシステムのユーザに配布して、該ユーザが支払う対価を受領し、

前記マルチプロセッサシステムは、前記コードを用いてセキュアタスクを実行するセキュアプロセッサと、通常タスクを実行する通常プロセッサを含み、該セキュアタスクと通常タスクの振り分けを行い、前記検証用情報を用いて該コードが正当であることを検証し、該コードを実行する

ことを特徴とするコード配布方法。

(付記 1 9) 前記コード作成者は、前記コードを 2 つ以上の部分に分割し、最初に一部を配布し、その後、前記ユーザの要求に応じて、残りの部分の対価を提示し、コードを提供して対価を受領することを特徴とする付記 1 8 記載のコード配布方法。

【 0 0 8 8 】

【発明の効果】

本発明によれば、セキュアプロセッサを含むヘテロなマルチプロセッサシステムを構築して、セキュアタスクと非セキュアタスクを振り分けることで、OS による制御が簡単になり、セキュア処理を効率よく実行することができる。また、セキュアタスクのコードに署名を施す際に、メモリに読み込む部分毎に署名を付与することで、コードを効率よく実行することが可能となる。

【図面の簡単な説明】

【図 1】

本発明のコード実行装置の原理図である。

【図 2】

第 1 のマルチプロセッサシステムの構成図である。

【図 3】

セキュアメモリとセキュアプロセッサの第 1 の構成図である。

【図 4】

セキュアプロセッサの構成図である。

【図 5】

セキュア OS の構成図である。

【図 6】

セキュアタスク管理の処理のフローチャートである。

【図 7】

セキュアメモリ管理の処理のフローチャートである。

【図 8】

実メモリ解放処理のフローチャートである。

【図 9】

セキュアファイルシステムの処理のフローチャートである。

【図 1 0】

セキュアメモリとセキュアプロセッサの第 2 の構成図である。

【図 1 1】

セキュアドライブ／メディアとセキュアメモリの構成図である。

【図 1 2】

第 2 のマルチプロセッサシステムの構成図である。

【図 1 3】

第 3 のマルチプロセッサシステムの構成図である。

【図 1 4】

回路生成処理のフローチャートである。

【図 1 5】

基本回路の配列を示す図である。

【図 1 6】

演算器群を示す図である。

【図 1 7】

第 1 のコード配布方法を示す図である。

【図 1 8】

第 1 の対価の支払いを示す図である。

【図 1 9】

第 2 のコード配布方法を示す図である。

【図 2 0】

第 3 のコード配布方法を示す図である。

【図 2 1】

第 2 の対価の支払いを示す図である。

【図 2 2】

第 4 のコード配布方法を示す図である。

【図 2 3】

記録媒体を示す図である。

【符号の説明】

- 1 1 セキュアメモリ手段
- 1 2 セキュアプロセッサ手段
- 1 3 通常メモリ手段
- 1 4 通常プロセッサ手段
- 1 5 制御手段
- 2 1 通常メモリ
- 2 2 通常プロセッサ
- 2 3、9 1 セキュアメモリ
- 2 4 セキュアプロセッサ
- 2 5 セキュアドライブ／メディア
- 2 6、9 2、9 4、9 5、9 6 システムバス
- 3 1 認証局公開鍵
- 3 2 署名検証部
- 3 3 署名保持部
- 3 4 ページ
- 4 1 復号鍵設定部
- 4 2 復号鍵保持部
- 4 3、6 3、7 2、8 4 復号部
- 4 4 プロセッサ
- 4 5 復号鍵指示部
- 5 1 セキュア OS
- 5 2 セキュアタスク管理
- 5 3 セキュアメモリ管理

5 4 セキュアファイルシステム

6 1、7 1、8 3 相互認証／セッション鍵共有部

6 2、8 5 暗号部

8 1 格納媒体

8 2 格納装置固有鍵

8 6 格納媒体固有鍵

8 7 暗号化コード

9 3 共有メモリ

1 0 1 コード作成者

1 0 2 コード認証機関

1 0 3 コード利用者

1 1 1 サーバ

1 1 2 データベース

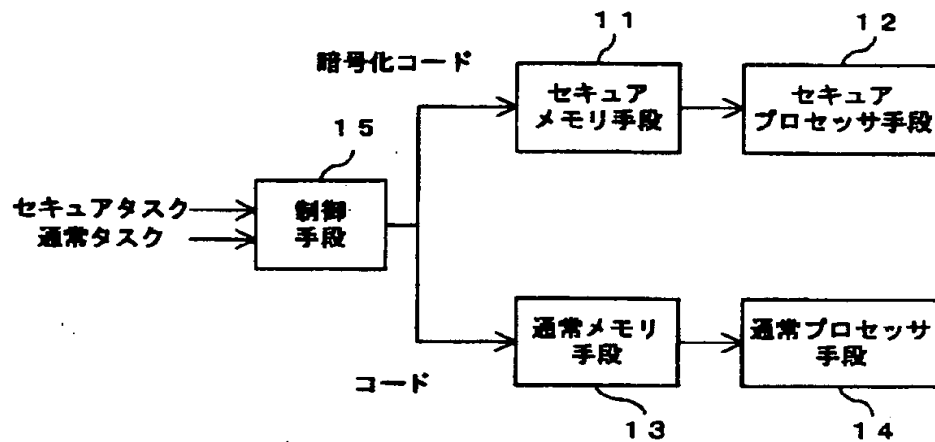
1 1 3 可搬記録媒体

1 1 4 メモリ

【書類名】 図面

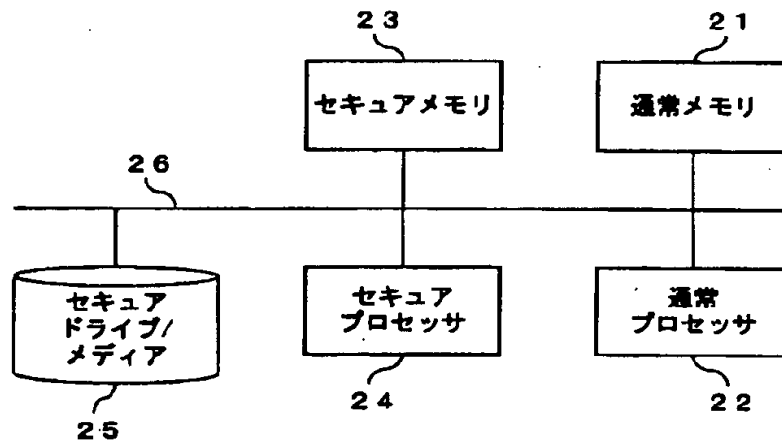
【図 1】

本 発 明 の 原 理 図



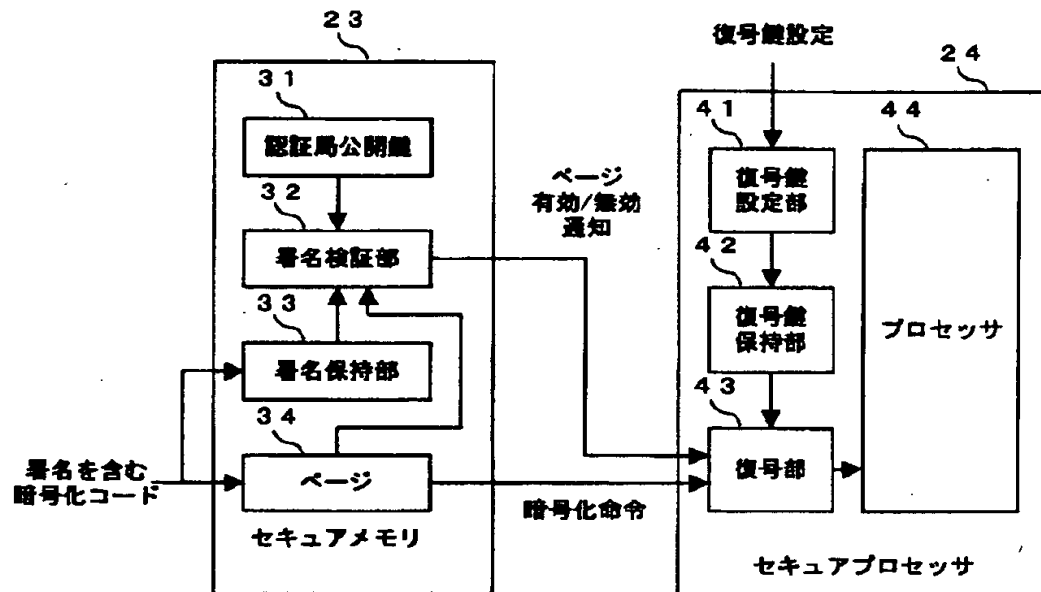
【図 2】

第 1 のマルチプロセッサシステムの構成図



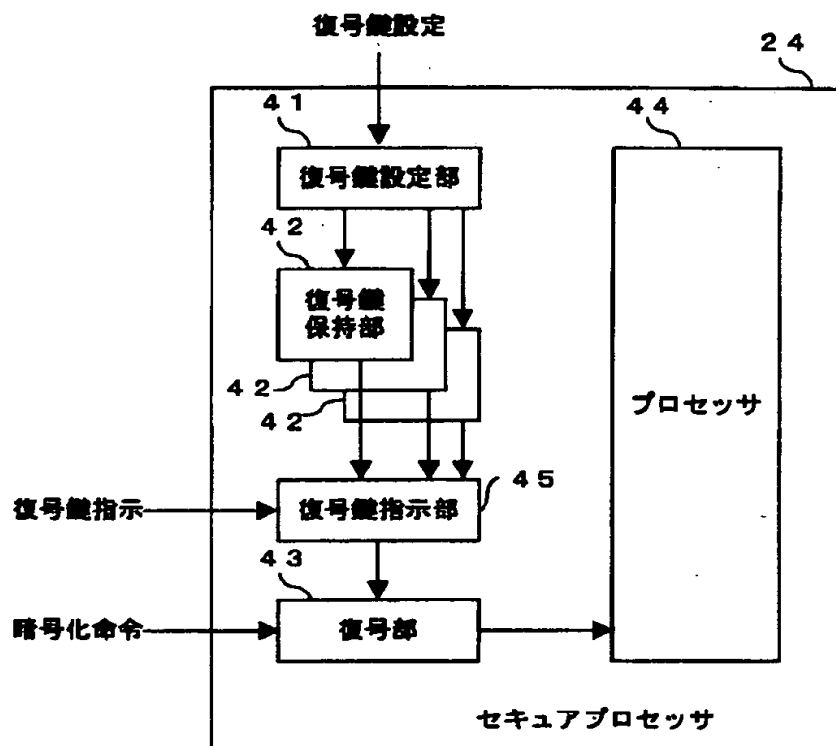
【図 3】

セキュアメモリとセキュアプロセッサの第1の構成図



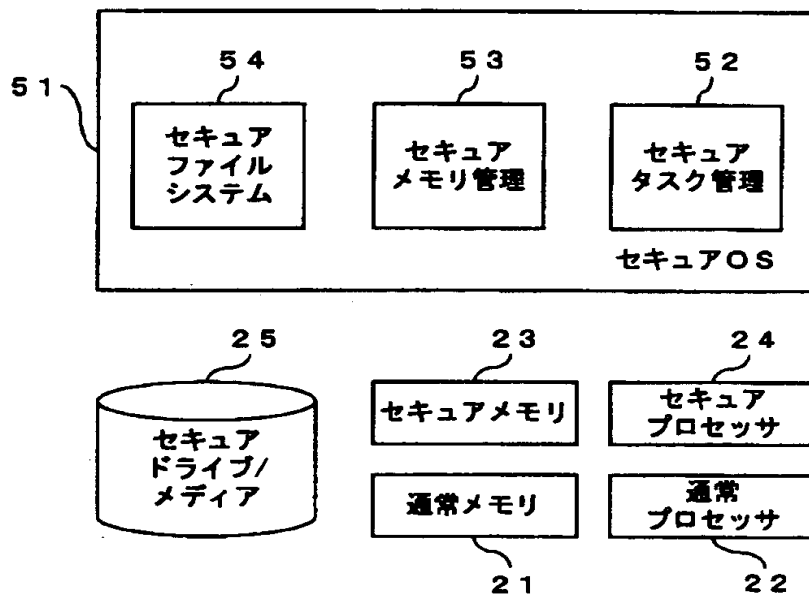
【図 4】

セキュアプロセッサの構成図



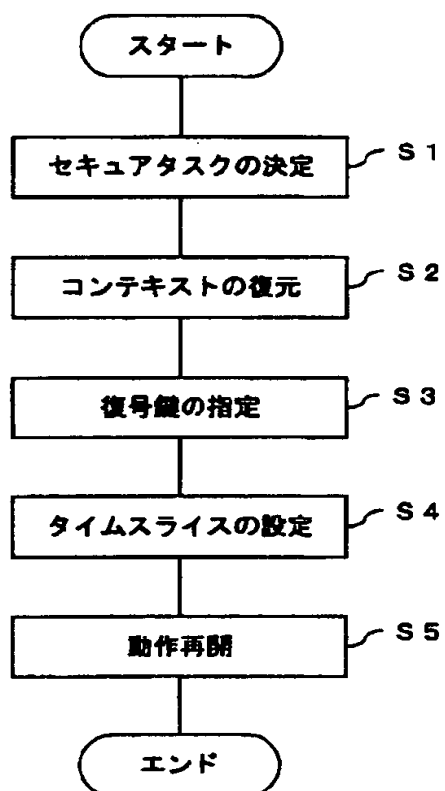
【図5】

セキュアOSの構成図



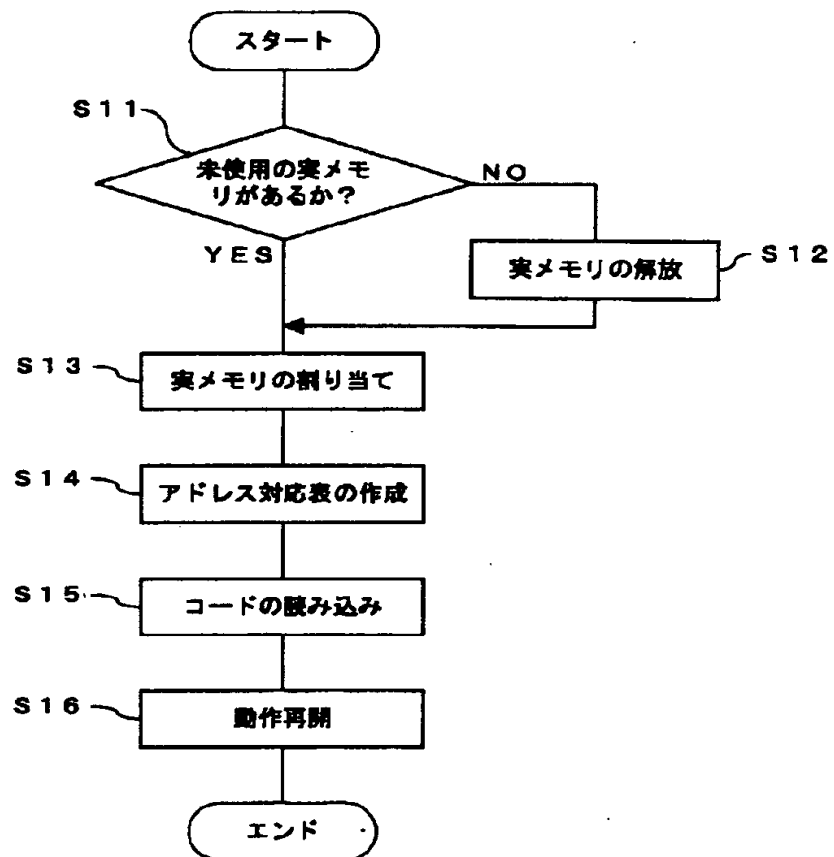
【図 6】

セキュアタスク管理の処理へのフローチャート



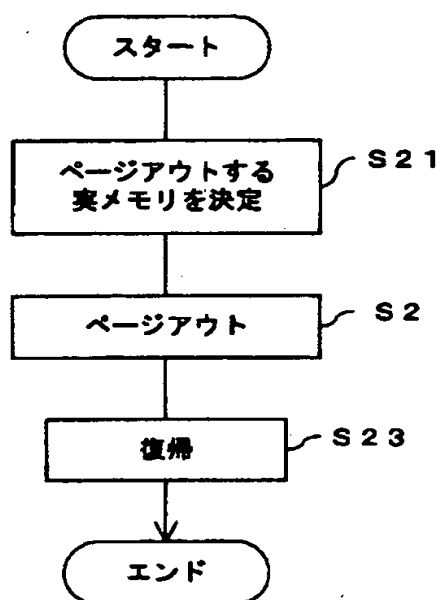
【図 7】

セキュアメモリ管理の処理のフローチャート



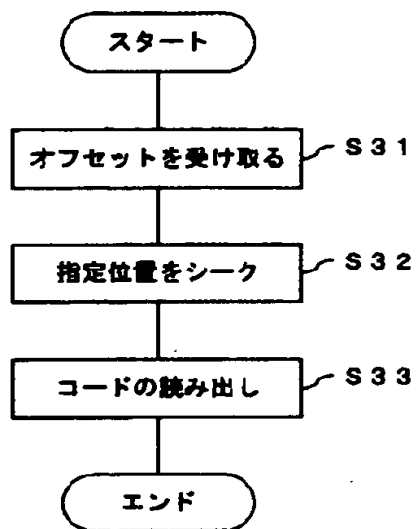
【図 8】

実メモリ解放処理のフローチャート



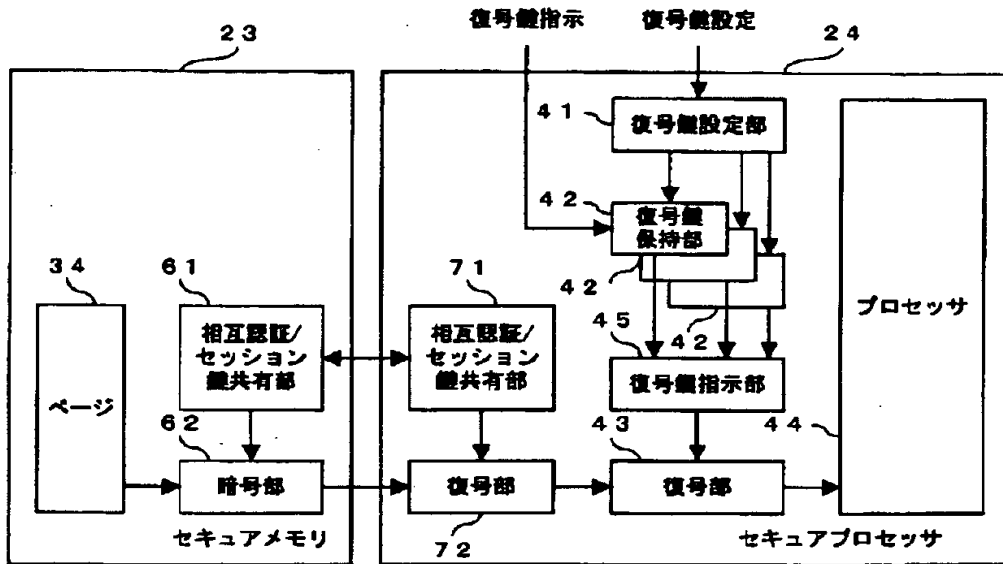
【図 9】

セキュアファイルシステムの処理のフローチャート



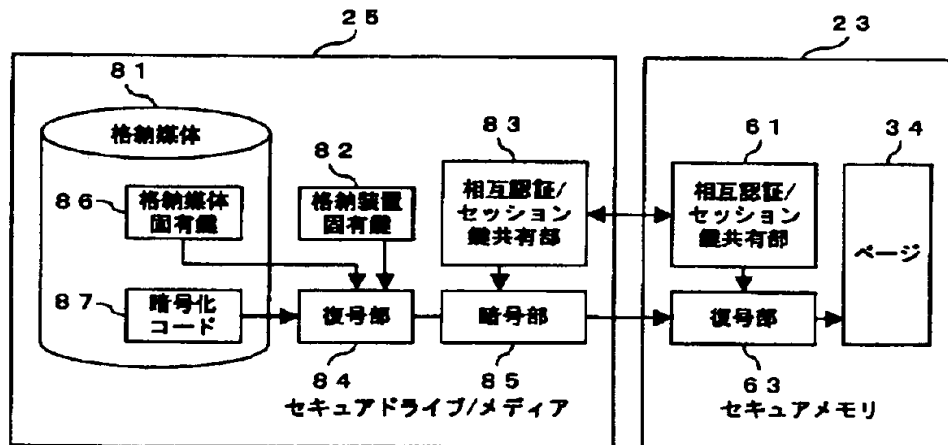
【図 1 0】

セキュアメモリとセキュアプロセッサの第 2 の構成図



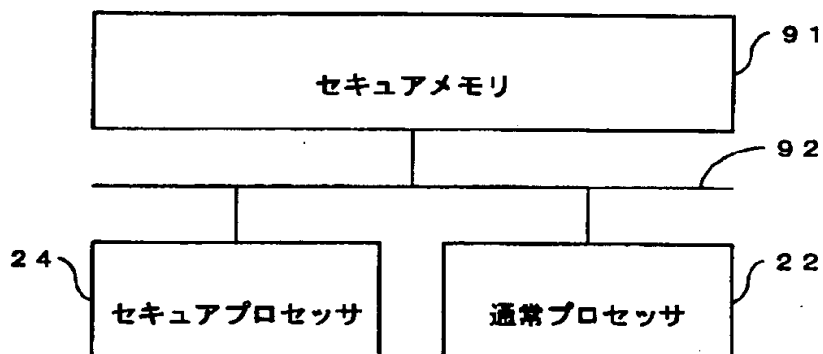
【図 11】

セキュアドライブ/メディアとセキュアメモリの構成図



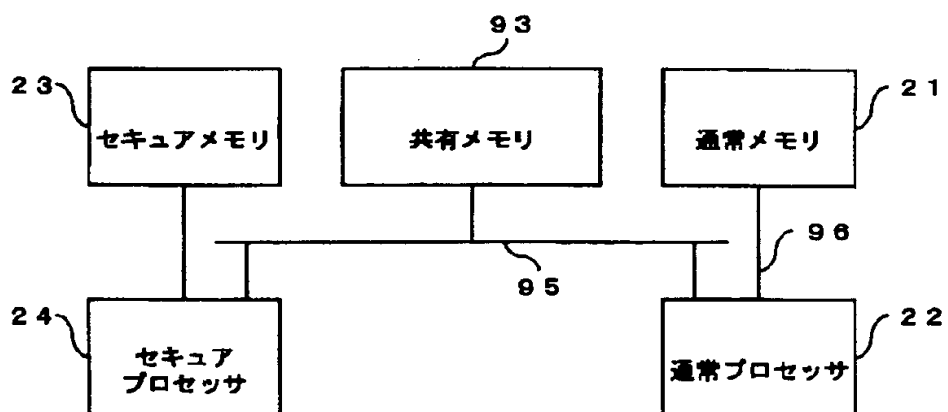
【図 12】

第2のマルチプロセッサシステムの構成図



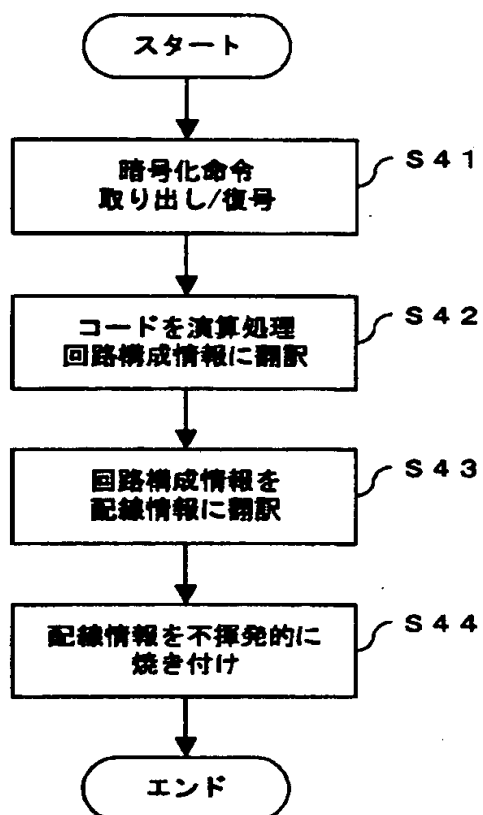
【図13】

第3のマルチプロセッサシステムの構成図



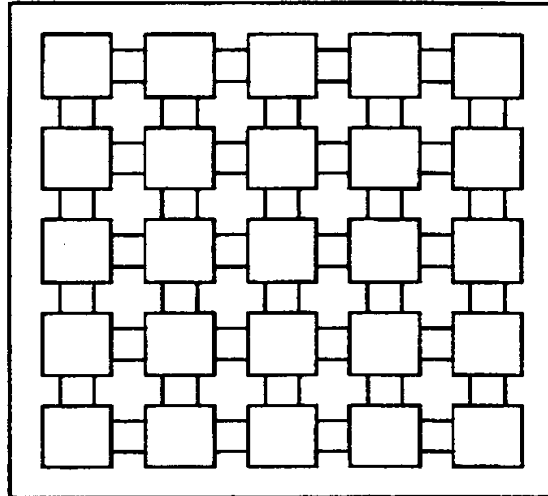
【図 1 4】

回路生成処理のフローチャート



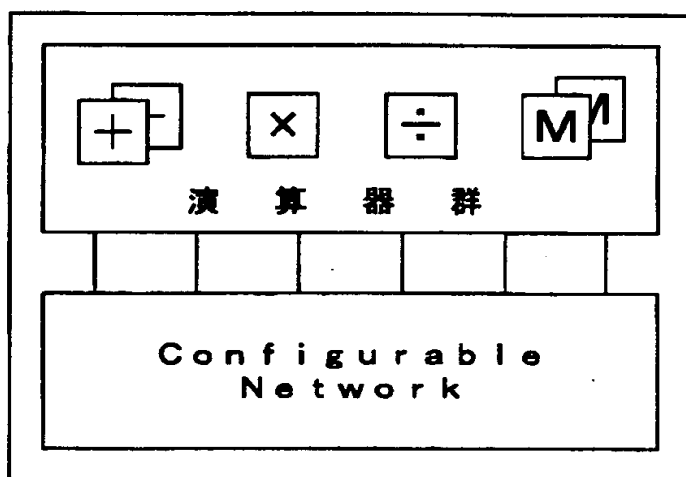
【図 1 5】

基本回路の配列を示す図



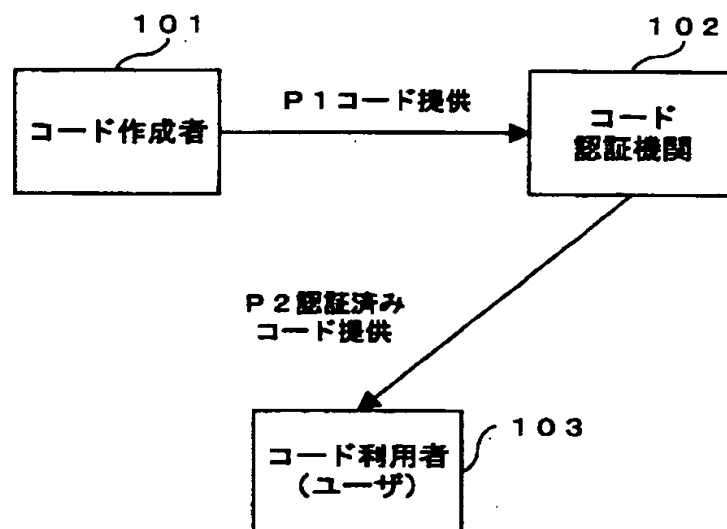
【図 1 6】

演算器群を示す図



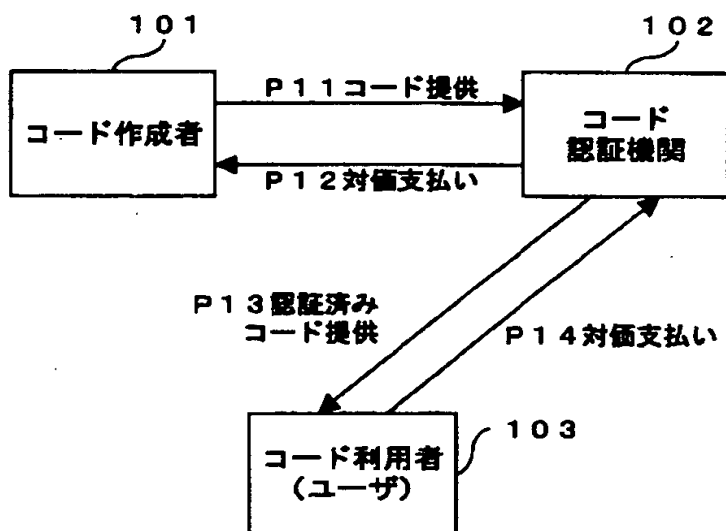
【図 1 7】

第 1 のコード配布方法を示す図



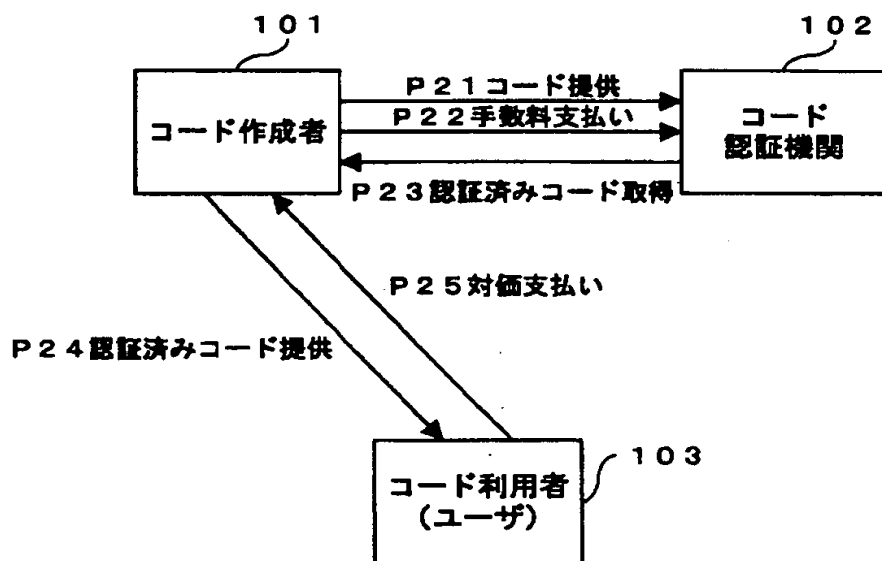
【図 1 8】

第 1 の対価の支払いを示す図



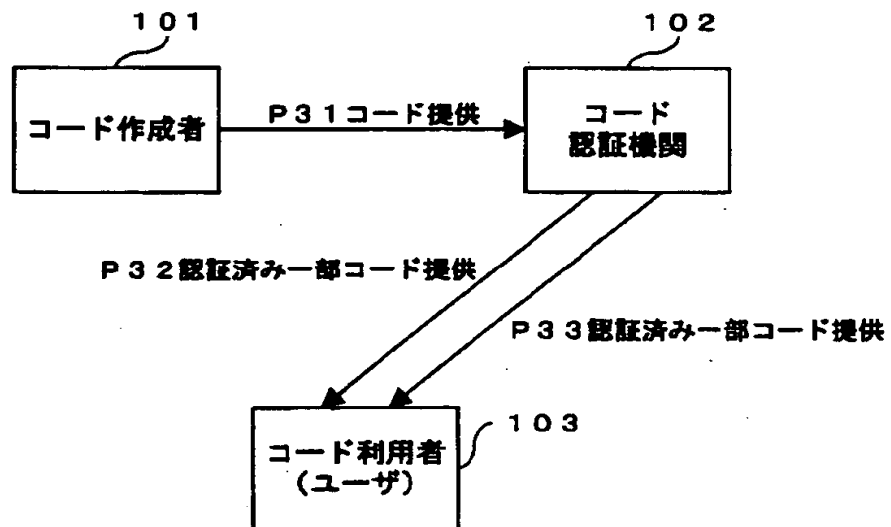
【図 1 9】

第 2 のコード配布方法を示す図



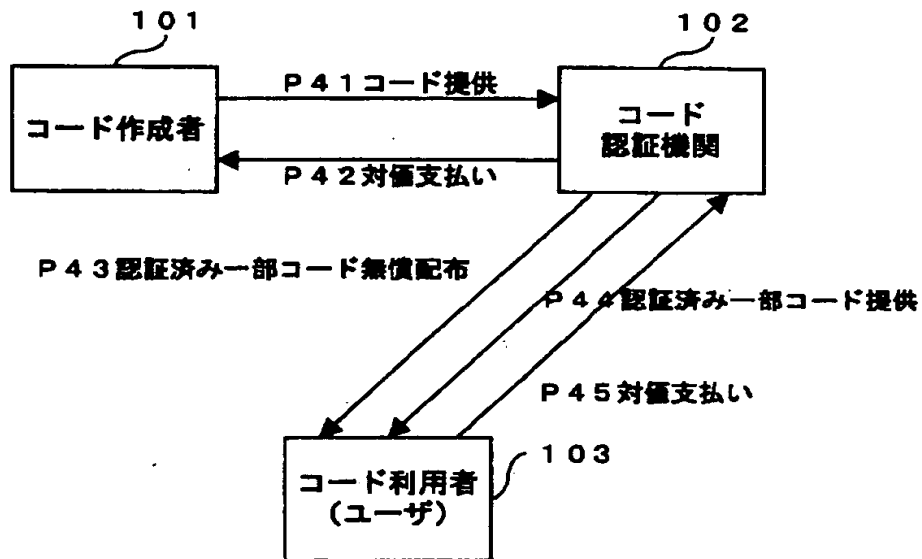
【図 2 0】

第 3 のコード配布方法を示す図



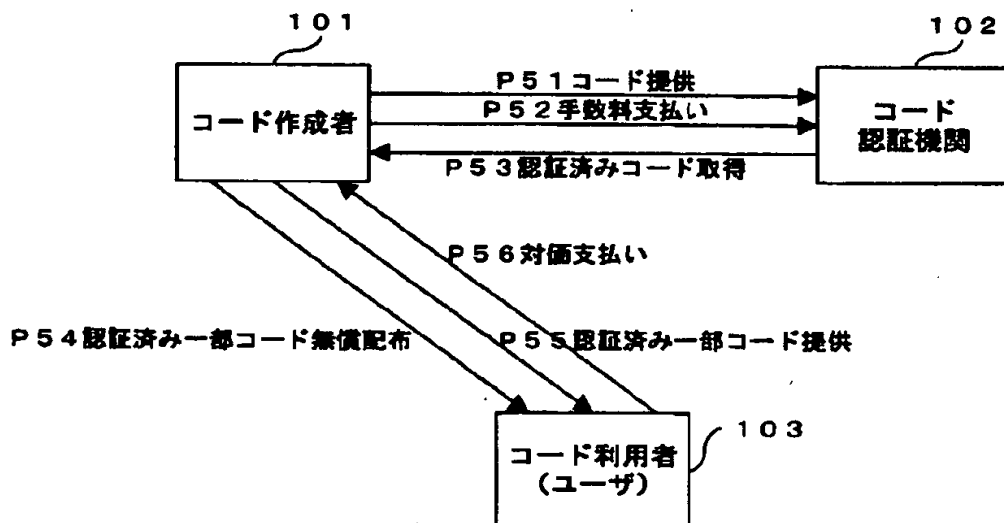
【図 2 1】

第 2 の対価の支払いを示す図



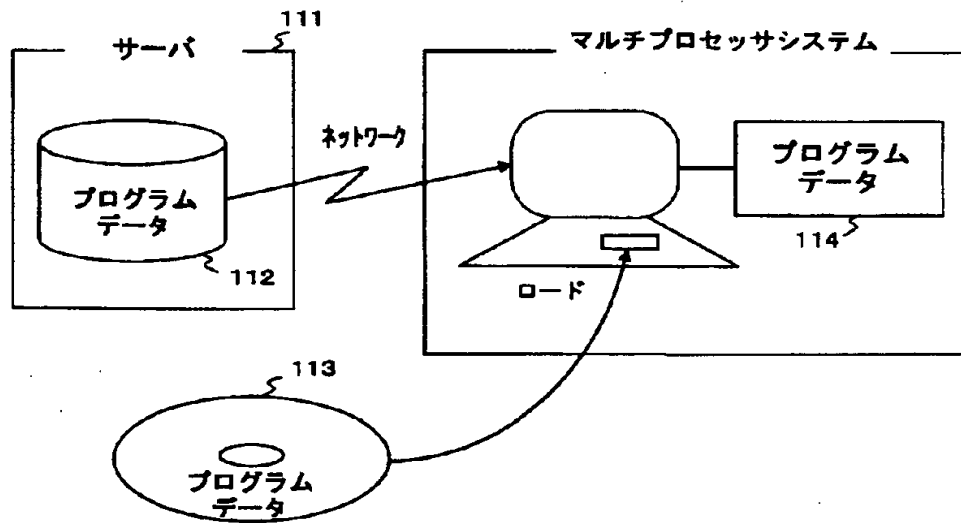
【図 2 2】

第 4 のコード配布方法を示す図



【図 2 3】

記 録 媒 体 を 示 す 図



【書類名】 要約書

【要約】

【課題】 既存OSを大幅に変更することなく、電子署名と暗号化が施されたコードを効率よく実行することが課題である。

【解決手段】 セキュアプロセッサ24と通常プロセッサ22を含むヘテロなマルチプロセッサシステムを構築し、セキュアタスクと非セキュアタスクを各プロセッサへ振り分ける。セキュアタスクの暗号化コードは、セキュアメモリ23に格納され、セキュアメモリ23は、認証局の公開鍵で署名を検証して、暗号化コードが有効であることをセキュアプロセッサ24に通知する。セキュアプロセッサ24は、セキュアメモリ23から暗号化命令をフェッチし、復号して実行する。

【選択図】 図2

出 願 人 履 歴 情 報

識別番号 [000005223]

1. 変更年月日 1996年 3月26日

[変更理由] 住所変更

住 所 神奈川県川崎市中原区上小田中4丁目1番1号

氏 名 富士通株式会社